Maximum Nash Welfare and Other Stories About EFX*

Georgios Amanatidis[1,2,3], Georgios Birmpas[4], Aris Filos-Ratsikas[5],
Alexandros Hollender[4], and Alexandros A. Voudouris[†1]

[1]University of Essex, U.K.
[2]University of Amsterdam, Netherlands
[3]Sapienza University of Rome, Italy
[4]University of Oxford, U.K.
[5]University of Liverpool, U.K.

Abstract

We consider the classic problem of fairly allocating indivisible goods among agents with additive valuation functions and explore the connection between two prominent fairness notions: maximum Nash welfare (MNW) and envy-freeness up to any good (EFX). We establish that an MNW allocation is always EFX as long as there are at most two possible values for the goods, whereas this implication is no longer true for three or more distinct values. As a notable consequence, this proves the existence of EFX allocations for these restricted valuation functions. While the efficient computation of an MNW allocation for two possible values remains an open problem, we present a novel algorithm for directly constructing EFX allocations in this setting. Finally, we study the question of whether an MNW allocation implies any EFX guarantee for general additive valuation functions under a natural new interpretation of approximate EFX allocations.

Keywords: Fair division; Nash Welfare; EFX; Approximation

1 Introduction

Fair division refers to the general problem of allocating a set of resources to a set of agents in a way satisfying a desired fairness criterion. A well-known example of such a criterion is *envy-freeness* [Gamow

* A preliminary version of this paper appeared in *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI)*. Email addresses: georgios.amanatidis@essex.ac.uk; georgios.birmpas@cs.ox.ac.uk; aris.filos-ratsikas@liverpool.ac.uk; alexandros.hollender@cs.ox.ac.uk; alexandros.voudouris@essex.ac.uk.

†Corresponding author.

and Stern, 1958, Foley, 1967, Varian, 1974], where each agent perceives the share she receives to be no worse than what any other agent receives. Since the problem was formally introduced by Banach, Knaster and Steinhaus [Steinhaus, 1948], fair division has attracted the attention of various scientific disciplines, including mathematics, economics, and political science. During the last two decades, the algorithmic aspects of fair division have been the focus of a particularly active line of work within the computer science community, e.g., see [Procaccia, 2016, Bouveret et al., 2016, Markakis, 2017] and references therein.

We consider the classic setting where the resources are indivisible goods that need to be fully allocated and the agents have additive valuation functions. One of the main challenges in this setting is that classic fairness notions such as equitability, envy-freeness and proportionality—introduced several decades ago having divisible resources in mind—are impossible to satisfy. To see this for envy-freeness, it suffices to consider two agents and one good of value; the agent who does not get the good is going to be envious. This has led to the recent emergence of several weaker fairness notions (see Related work). As a result, there is a plethora of open questions about the existence, the computation and the interrelationships of such notions. In this work we focus on two of the most prominent: envy-freeness up to any good (EFX) and maximum Nash welfare (MNW).

EFX, introduced recently by Gourvès et al. [2014] and Caragiannis et al. [2019b], is an additive relaxation of envy-freeness. Here an agent may envy another agent but only by the value of the least desirable good in the other agent's bundle. While this added flexibility of EFX takes care of extreme pathological cases like the one mentioned above (2 agents, 1 good), this notion is not well understood yet. Despite the active interest in it, it is not known whether EFX allocations always exist, even for 4 agents with additive valuation functions.[1] We consider the problem of showing the existence of EFX allocations to be one of the most intriguing currently open questions in fair division.

The *Nash social welfare* (or, simply, Nash welfare) is the geometric mean of the agents' utilities. By considering maximum Nash welfare (MNW) allocations, i.e., allocations that maximize the product of the utilities, we achieve some kind of balance between the *efficiency* of the maximum utilitarian social welfare—the sum of the utilities—and the *individual fairness* of the maximum egalitarian social welfare—the minimum utility. Although not a fairness concept per se, MNW has strong ties to fairness. In the setting where the goods are divisible, each (possibly fractional) MNW allocation corresponds to a *competitive equilibrium from equal incomes*, a market equilibrium (under the assumption that all agents are endowed with the same budget) that is known to guarantee envy-freeness and Pareto optimality [Varian, 1974]. Even in our setting, Caragiannis et al. [2019b] showed that integral MNW allocations, besides being Pareto optimal, are *envy-free up to one good* (EF1) and approximately satisfy *maximin share fairness* up to

[1] The existence of EFX allocations for 2 agents was shown independently by Gourvès et al. [2014] and Plaut and Roughgarden [2018], while, very recently, Chaudhury et al. [2020a] presented an algorithm that computes an EFX allocation for instances with 3 agents.

a $\Theta(1/\sqrt{n})$ factor, where n is the number of agents. Both these guarantees are significantly weaker than EFX, in the sense that they are both implied by EFX but they do not imply *any* approximation of it.

In general, MNW does not imply EFX. One of our goals is to identify the cases where it does, in terms of the allowed number of distinct values for the goods. For such cases, we immediately obtain that EFX allocations must exist and then investigate how to efficiently compute them, either through maximizing the Nash welfare or directly. Since, in general, MNW does not even imply a non-trivial approximation of EFX, we further introduce a less stringent, yet natural, new interpretation of approximate EFX and investigate how it is related to MNW.

1.1 Our contribution

There are two variants of EFX used in the related literature, depending on whether only the *positively* valued goods are considered or not; for the latter case we adopt the name EFX_0 suggested by Kyropoulou et al. [2020]. We start by establishing a strong algorithmic connection between the two variants (Proposition 2.3). Then we explore the relationship between maximizing the Nash welfare and achieving EFX or EFX_0 allocations. In doing so, we also obtain some interesting results for the individual notions. In particular:

- In case there are at most two possible values for the goods (*2-value instances*; see the formal definition in Section 2), we show that any allocation that maximizes the Nash welfare is EFX_0 (Theorem 3.2). This has the following two consequences:

  - For any 2-value instance, there exists an EFX_0 allocation. Note that this is the first such existence result for non-identical valuations that holds for any number of agents and goods.
  - For the special case of binary valuations, by adapting an algorithm of Barman et al. [2018b], we can efficiently construct an allocation that is both MNW and EFX_0.

  Note that the implication MNW $\Rightarrow$ EFX_0 is no longer true for three or more distinct values.

- While for general 2-value instances the efficient computation of an MNW allocation remains an open problem, we propose a polynomial-time algorithm for producing EFX_0 allocations in this case (Theorem 4.1). This algorithm, which we call Match&Freeze, is based on repeatedly computing maximum matchings and "freezing" certain agents whenever they acquire too much value compared to their peers. We believe these novel ideas might be a stepping stone for proving the existence of EFX allocations in more general settings.

- We also show that the difficulty of computing EFX allocations does not depend solely on the different number of values, but also on the ratio between the maximum and the minimum value. In particular, for instances where the values of the agents lie in an interval such that the ratio between the maximum and the minimum value is at most 2, we can compute an EFX allocation using a simple variation of the well-known round-robin algorithm (Theorem 4.4).

- For general additive valuations, we show that an MNW allocation does not guarantee any non-trivial approximation of EFX. However, we argue that the current definition of *approximate* EFX allocations is not always meaningful. Instead, we explore a different natural definition based on the idea of (hypothetically) augmenting an agent's bundle until an EFX-like condition is satisfied. For this new benchmark, which we call EFX-*value*, we show that any MNW allocation is a $1/2$-approximation of EFX (Theorem 5.5).

1.2 Related work

As there is a vast literature on fair division, here we focus on the indivisible items setting and on related fairness notions. The concept of *envy-freeness up to one good* (EF1) was implicitly suggested by Lipton et al. [2004] and formally defined by Budish [2011]. Budish [2011] also introduced the notion of *maximin share* (MMS), which has been studied extensively [Kurokawa et al., 2018, Amanatidis et al., 2017, Barman and Murthy, 2017, Garg et al., 2019, Ghodsi et al., 2018, Garg and Taki, 2020] and has yielded several very interesting variants like *pairwise* MMS Caragiannis et al. [2019b], *groupwise* MMS [Barman et al., 2018a], and MMS for groups of agents Suksompong [2018].

As already mentioned, EFX was introduced by Gourvès et al. [2014] (under the term *near envy-freeness*) and popularized by Caragiannis et al. [2019b]. Plaut and Roughgarden [2018] defined the notion of α-approximate EFX (or α-EFX) allocations and studied exact and approximate EFX allocations with both additive and general valuations. Most of their results, including the existence of EFX allocations for identical valuations, hold under the similar but stricter notion of EFX_0 which is implicitly introduced therein. The currently best 0.618-approximation of either EFX or EFX_0 for the additive case is due to Amanatidis et al. [2020]. For binary additive valuations, Aleksandrov and Walsh [2019] recently proposed an algorithm that produces EFX—but not necessarily EFX_0—allocations. Independently and at the same time with our work, Babaioff et al. [2020] designed an algorithm that computes an EFX_0 allocation which maximizes the Nash welfare for submodular dichotomous valuations, a class that includes binary, but does not include general 2-value additive valuations.

Very recently, Manurangsi and Suksompong [2020] showed that EFX allocations exist with high probability for any number of agents and items under the assumption that the valuations of the agents are drawn at random from a probability distribution. The challenge of showing the existence of EFX allocations is nicely demonstrated in the recent work of Suksompong [2020], who showed that in instances with two agents there can be as few as two EFX allocations, while the number of EF1 allocations is always exponential in the number of items.

Besides [Caragiannis et al., 2019b], there are several recent papers which relate allocations that maximize (exactly or approximately) the Nash welfare with other fairness notions. Caragiannis et al. [2019a] showed that there exist *incomplete* allocations that are EFX and in which each agent receives at least half of the value they get in an MNW allocation; Chaudhury et al. [2020b] achieved the same with only a few unallocated goods. Garg and McGlaughlin [2019] showed how to get an allocation that 2-approximates the Nash welfare of an MNW allocation that is also *proportional up to one good*, satisfies a weak MMS guarantee and is Pareto optimal.

Since computing MNW allocations is an APX-hard problem [Lee, 2017], there is an active interest on special cases or on approximation algorithms. Barman et al. [2018b] show how to efficiently compute MNW allocations for binary additive valuation functions. Cole and Gkatzelis [2018] were the first to obtain a constant approximation algorithm to the MNW objective. This algorithm, as shown via the improved analysis of Cole et al. [2017], achieves a factor of 2. The currently best-known factor of 1.45 is due to Barman et al. [2018c]. Going beyond the additive case, in a recent work Garg et al. [2020] study the problem for submodular valuation functions.

2 Preliminaries and Notation

We consider fair division instances $I = (N, M, (v_i)_{i \in N})$ in which there is a set N of n *agents* and a set M of m indivisible *goods*. Each agent $i \in N$ has a *valuation function* $v_i : M \to \mathbb{R}_{\geq 0}$ assigning a non-negative real value $v_i(g)$ to each good $g \in M$. Throughout this work, v_i is *additive*, i.e., $v_i(A) = \sum_{g \in A} v_i(g)$ for every set (or *bundle*) of goods $A \subseteq M$. We pay particular attention to the following subclasses of additive valuation functions:

- *Binary*: $v_i(g) \in \{0, 1\}$ for every $i \in N$ and $g \in M$;
- *k-value*: there is a set V consisting of $|V| = k$ distinct, non-negative real values such that $v_i(g) \in V$ for every $i \in N$ and $g \in M$;
- *Interval-value*: for every agent $i \in N$ there exist two real non-negative numbers x_i and y_i such that $x_i < y_i$, and $v_i(g) \in [x_i, y_i]$ for every $g \in M$.

Of course, any binary instance is a 2-value instance with $V = \{0, 1\}$, but we distinguish between these cases as we are able to obtain stronger algorithmic results for the binary case.

A *complete allocation* (or just *allocation*) $\mathbf{A} = (A_i)_{i \in N}$ is a vector listing the bundle A_i of goods that each agent i receives, such that $A_i \cap A_j = \varnothing$ for every $i, j \in N$, and $\cup_{i \in N} A_i = M$. Our goal is to come up with allocations that are considered to be *fair* by all agents. We begin by defining envy-freeness and its additive relaxations.

Definition 2.1. An allocation $\mathbf{A} = (A_i)_{i \in N}$ is

- *envy-free* (EF) if $v_i(A_i) \geq v_i(A_j)$ for every pair $i, j \in N$;
- *envy-free up to one good* (EF1) if for every pair $i, j \in N$ with $A_j \neq \varnothing$ there exists a good $g \in A_j$, such that $v_i(A_i) \geq v_i(A_j \setminus \{g\})$;
- *envy-free up to any (positively-valued) good* (EFX) if for every pair $i, j \in N$ and every good $g \in A_j$ for which $v_i(g) > 0$, it holds that $v_i(A_i) \geq v_i(A_j \setminus \{g\})$;
- *envy-free up to any good* (EFX_0) if for every pair $i, j \in N$ and every good $g \in A_j$, it holds that $v_i(A_i) \geq v_i(A_j \setminus \{g\})$.

By definition, we have EF $\Rightarrow$ EFX_0 $\Rightarrow$ EFX $\Rightarrow$ EF1. In words, any EF allocation is also EFX_0, any EFX_0 allocation is also EFX, and any EFX allocation is also EF1. However, none of these implications work in the opposite direction. For brevity, we say that agent i is $\mathcal{E}$ towards agent j when the criterion of $\mathcal{E} \in \{\text{EF}, \text{EF1}, \text{EFX}, \text{EFX}_0\}$ is true for the ordered pair (i, j).

As mentioned in the Introduction, the Nash welfare is usually defined as the geometric mean of the values. Here, for simplicity, we use the *product* of the values instead. As the allocations (exactly) maximizing the Nash welfare are the same under both definitions, this is without loss of generality.

Definition 2.2. The *Nash welfare* of an allocation $\mathbf{A} = (A_i)_{i \in N}$ is the product of the values of the agents for their bundles: $\text{NW}(\mathbf{A}) = \prod_{i \in N} v_i(A_i)$.

We will usually denote by $\mathbf{A}^*$ one of the allocations that maximize the Nash welfare (MNW). Among all such allocations, we will sometimes select $\mathbf{A}^*$ so that some additional properties are satisfied; e.g., see the discussion in Section 3. Caragiannis et al. [2019b] showed that MNW $\Rightarrow$ EF1, but the exact connection between MNW and the variants of EFX is not well-understood.

Before we dive into our main technical results, we show a somewhat surprising connection between EFX and EFX_0. In particular, assuming agents with additive valuation functions, the question of finding an EFX_0 allocation reduces to finding an EFX allocation for an additive instance with only slightly perturbed valuation functions (and the same number of distinct values). An immediate corollary is that the existence (resp. the efficient computation) of EFX allocations for additive agents implies the existence (resp. the efficient computation) of EFX_0 allocations; the converse statements are obvious.

Proposition 2.3. *The problem of computing* EFX_0 *allocations for additive instances reduces to the problem of computing* EFX *allocations for additive instances with the same number of distinct values for the goods. When all values are rational numbers, this reduction requires only polynomial time.*

Proof. Consider any instance $I = (N, M, (v_i)_{i\in N})$. Let δ be the minimum non-zero value difference among any two subsets of goods, according to the valuation function of any agent, that is

$$\delta = \min_{i\in N} \min_{\substack{A,B\subseteq 2^M:\\ v_i(A)<v_i(B)}} \{v_i(B) - v_i(A)\}.$$

Furthermore, pick an arbitrary $\varepsilon \in (0, \frac{\delta}{m})$. Now, let $I' = (N, M, (\tilde{v}_i)_{i\in N})$ be an instance such that

$$\tilde{v}_i(g) = \begin{cases} v_i(g), & \text{if } v_i(g) > 0 \\ \varepsilon, & \text{if } v_i(g) = 0. \end{cases}$$

That is, I' is obtained from I by changing any 0 in the valuation functions of the agents to ε. Assume that there exists an EFX allocation $\mathbf{A}$ for instance I'. We will show that $\mathbf{A}$ is also an EFX_0 allocation for I.

Consider any pair of agents $i, j \in N$. Since $\mathbf{A}$ is EFX in I', we have that $\tilde{v}_i(A_i) \geq \tilde{v}_i(A_j \setminus \{g\})$ for every $g \in A_j$. Let $g^* = \arg\min_{g\in A_j} v_i(g)$. Observe that by the choice of ε, g^* is the one that must be ignored when we check whether $\mathbf{A}$ is EFX_0 for I as well.

- If $v_i(g^*) = 0$, then it must be the case that $v_i(A_i) \geq v_i(A_j)$. Assume otherwise that $v_i(A_i) < v_i(A_j)$. Then, by the definition of δ, it must be $v_i(A_i) \leq v_i(A_j) - \delta$. Using the fact that $v_i(A_j) = v_i(A_j \setminus \{g^*\}) \leq \tilde{v}_i(A_j \setminus \{g^*\})$, by our choice of ε, we have that

  $$\tilde{v}_i(A_i) \leq v_i(A_i) + m\varepsilon \leq v_i(A_j) - \delta + m\varepsilon < \tilde{v}_i(A_j \setminus \{g^*\}),$$

  contradicting the assumption that $\mathbf{A}$ is EFX in I'. Hence, i is envy-free towards j.

- If $v_i(g^*) > 0$, then it must be the case that $v_i(A_i) \geq v_i(A_j \setminus \{g^*\})$. As before, assume otherwise that $v_i(A_i) < v_i(A_j \setminus \{g^*\})$ or, equivalently, $v_i(A_i) \leq v_i(A_j \setminus \{g^*\}) - \delta$. Since $v_i(A_j \setminus \{g^*\}) \leq \tilde{v}_i(A_j \setminus \{g^*\})$, and by our choice of ε, we have that

  $$\tilde{v}_i(A_i) \leq v_i(A_i) + m\varepsilon \leq \tilde{v}_i(A_j \setminus \{g^*\}) - \delta + m\varepsilon < \tilde{v}_i(A_j \setminus \{g^*\}),$$

  again contradicting the assumption that $\mathbf{A}$ is EFX in I'. Hence, i is EFX_0 towards j.

Therefore the computation of an EFX_0 allocation can be reduced to computing an EFX allocation in an instance with slightly perturbed valuation functions as above.

In case the values are rational numbers, this reduction needs only polynomial time as δ is at least $1/D$, where D is the denominator of the product of the values of all agents for all goods, and hence it suffices to choose $\varepsilon \in (0, \frac{1}{mD})$. □

3 Maximum Nash Welfare: EFX and Computational Complexity

In this section we focus on allocations that maximize the Nash welfare. We first identify the subclasses of valuation functions for which the MNW allocations are always EFX_0, and then consider computational complexity questions.

Before moving forward, we need to discuss how we handle instances with zero Nash welfare and instances containing *zero-valued* goods, i.e., goods for which all agents have value 0. One obvious solution for zero-valued goods is to just discard them. However, we show that it is possible to handle them even in settings where one cannot assume *free disposal* and all goods must be allocated; see the paper of Bei et al. [2020] for examples of such settings.

Instances with zero Nash welfare. When we talk about the MNW allocations of an instance, the standard interpretation would be to include *all* complete allocations which achieve the maximum Nash welfare. When it is possible to achieve positive Nash welfare this is indeed true. However, for the extreme case of instances where all allocations have zero Nash welfare we are going to need a refinement. Following the work of Caragiannis et al. [2019b], we call an allocation an MNW allocation if it (1) maximizes the cardinality of the set of agents with positive value, say S, and then (2) maximizes the product of the values of agents in S.

The requirements (1) and (2) are by default true for MNW allocations in instances with positive Nash welfare. They are also necessary because when the Nash welfare is zero, the idea of maximizing it clearly fails to distinguish "good" allocations in any sense. To illustrate this, consider the next instance:

	g_1	g_2	g_3
agent 1	1	0	0
agent 2	1	0	0
agent 3	0	1	1

Since the first two agents only like g_1, the Nash welfare of any allocation is 0. However, not all allocations are EFX_0. The allocation $\{\varnothing, \varnothing, \{g_1, g_2, g_3\}\}$ is clearly not EFX_0 since the first two agents envy agent 3 even after the removal of either g_2 or g_3. Even an allocation such as $\{\{g_1, g_2\}, \varnothing, \{g_3\}\}$, which maximizes the number of agents with positive value, is not EFX_0 since agent 2 envies agent 1 even after the removal of g_2. On the other hand, the allocation $\{\{g_1\}, \varnothing, \{g_2, g_3\}\}$, which maximizes the number of agents with positive value *as well as* the product of their values, is indeed EFX_0: the envy of agent 2 towards agent 1 is eliminated by the removal of g_1.

Instances with zero-valued goods. While, clearly, zero-valued goods do not affect the Nash welfare of an allocation, they do play an important role as to whether this allocation is EFX_0. To allocate such goods, we first ignore them completely, and compute a Nash welfare maximizing partial allocation $\mathbf{B}^*$ only for the remaining goods (which are positively valued by some agent), subject to the requirements (1) and (2) in case the Nash welfare is zero. We then obtain the complete MNW allocation $\mathbf{A}^*$ by allocating all the zero-valued goods to one of the agents with the least value according to $\mathbf{B}^*$.[2] Observe that $\mathrm{NW}(\mathbf{A}^*) = \mathrm{NW}(\mathbf{B}^*)$, by definition. Allocating the zero-valued goods this way is also necessary as we illustrate next. Consider the same example as above, but with an extra zero-valued good g_4, such that:

	g_1	g_2	g_3	g_4
agent 1	1	0	0	0
agent 2	1	0	0	0
agent 3	0	1	1	0

As before, all allocations have zero Nash welfare, and hence we need an allocation that satisfies (1) and (2). The allocation $(\{g_1, g_4\}, \varnothing, \{g_2, g_3\})$ is indeed such an allocation: the number of agents with positive value as well as the product of their values are maximized. However, because g_4 has been given to agent 1 (who has value 1) instead of 2 (who has value 0), agent 2 envies agent 1 even after g_4's removal from the bundle of agent 1, and thus the allocation is not EFX_0. By moving g_4 to agent 2, we obtain the allocation $(\{g_1\}, \{g_4\}, \{g_2, g_3\})$, which maximizes the number of agents with positive value, the product of their values, and gives the all-zero good g_4 to the agent with the least value among all agents, and is indeed EFX_0 as agent 1 has only one good.

3.1 When does MNW imply EFX?

Our main result here is that for all 2-value instances any MNW allocation is also EFX_0. Moreover, this result is tight: there exist 3-value instances for which this implication is no longer true. To simplify the presentation of our results, we distinguish between binary and general 2-value instances.

Theorem 3.1. *For every binary instance, any MNW allocation is* EFX_0.

Proof. Consider any binary instance $I = (N, M, (v_i)_{i \in N})$, and let $\mathbf{B}^* = (B_i)_{i \in N}$ be the allocation that maximizes the Nash welfare (by maximizing the number of agents with positive value and then the product of their value in case the MNW is zero) for the sub-instance $I_{>0}$ consisting only of the goods which are positively valued by some agent. Then, $\mathbf{A}^*$ is obtained from $\mathbf{B}^*$ by allocating the remaining goods (which are valued as zero by all agents) to one of the agents with the least value for their own bundles.

Observe that in $I_{>0}$, $\mathbf{B}^*$ must be such that all agents with positive value get goods which they value as 1 and all agents with zero value get an empty set. Assume otherwise that some agent i gets a good g

[2] For the restricted valuation classes we study here, this suffices. A more general alternative way to complete the allocation would be to allocate all the zero-valued goods to one of the agents that no one envies in $\mathbf{B}^*$. It is not hard to show that in any MNW (partial) allocation at least one such agent exists.

such that $v_i(g) = 0$. Then by moving g to some agent $j \neq i$ with $v_j(g) = 1$ we can strictly increase either the product of the values of the agents that have positive value or the number of agents that get positive value, a contradiction. Hence, we have that $v_i(B_i) = |B_i|$.

We next show that by allocating all the zero-valued goods to some agent $i^* \in \arg\min_{i \in N} v_i(B_i)$ we have that $\mathbf{A}^*$ is EFX$_0$ as long as $\mathbf{B}^*$ is EFX$_0$. We distinguish between two cases depending on whether $\mathrm{NW}(\mathbf{B}^*) > 0$ or $\mathrm{NW}(\mathbf{B}^*) = 0$.

Case I: $\mathrm{NW}(\mathbf{B}^*) > 0$.

Consider a pair of agents i and j. If $\min_{g \in B_j} v_i(g) = 1$, then i must be EFX$_0$ towards j since the notion of EFX$_0$ coincides with that of EF1 in this case, and any MNW allocation is EF1 [Caragiannis et al., 2019b] . So, from now on we assume that $\min_{g \in B_j} v_i(g) = 0$. Moreover, we assume that there exists a good in B_j that i values as 1, since otherwise i would trivially be EFX$_0$ towards j, and hence $|B_j| \geq 2$.

We will show that i is in fact envy-free towards j, and thus $v_i(B_i) \geq v_i(B_j)$. Assume towards a contradiction that $v_i(B_i) < v_i(B_j)$. Since there exists a good $g \in B_j$ such that $v_i(g) = 0$, we have that $v_i(B_j) < v_j(B_j)$. By the fact that $v_i(B_i) = |B_i|$ and $v_j(B_j) = |B_j|$, we thus obtain that $|B_j| \geq |B_i| + 2$. Now, define a new allocation by moving a good in B_j that i values as 1 from j to i. The product of the values of the two agents in the new allocation is equal to

$$(|B_i| + 1)(|B_j| - 1) = |B_i||B_j| + |B_j| - |B_i| - 1 \geq |B_i||B_j| + 1.$$

Since the bundles of the remaining agents have not changed, the new allocation has strictly higher Nash welfare compared to $\mathbf{B}^*$, a contradiction.

Case II: $\mathrm{NW}(\mathbf{B}^*) = 0$.

Consider a pair of agents i and j. If $B_i = \varnothing$ and $B_j = \varnothing$, then they are trivially envy-free towards each other. Also, if $B_i \neq \varnothing$ and $B_j \neq \varnothing$, we can show that they are EFX$_0$ towards each other by adapting our arguments for the previous case. Hence, we now focus on the case where $B_i = \varnothing$ and $B_j \neq \varnothing$. If $|B_j| = 1$, then i is trivially EFX$_0$ towards j. Hence, assume that $|B_j| \geq 2$. We claim that $\max_{g \in B_j} v_i(g) = 0$, and consequently i is envy-free towards j. Assume otherwise that there exists a good $g \in B_j$ such that $v_i(g) = 1$. Then, by moving g from j to i we can either obtain positive Nash welfare if i is the only agent with zero value, or we can increase the number of agents with positive value in case the Nash welfare remains equal to 0; since $v_j(B_j) = |B_j| \geq 2$, j still has positive value even after losing g.

In any case, we conclude that $\mathbf{B}^*$ is EFX$_0$, and consequently $\mathbf{A}^*$ is EFX$_0$ as well. $\square$

We continue by showing that maximizing the Nash welfare yields an EFX$_0$ allocation for all 2-value instances. Since a 2-value instance with values $a > 1$ and $b = 0$ is equivalent to a binary instance (by normalizing the values), Theorem 3.1 above implies that we only need to focus on instances with positive values. Note that in this case EFX$_0$ coincides with EFX.

Theorem 3.2. *For any* 2*-value instance with positive values, any MNW allocation is* EFX.

Proof. Let $a > b > 0$ and consider any 2-value instance $I = (N, M, (v_i)_{i \in N})$ in which $v_i(g) \in \{a, b\}$ for every $i \in N$ and $g \in M$. Let i and j be any two agents who are given the sets of goods A_i and A_j in an MNW allocation $\mathbf{A}^*$. We say that a good is of *type* T_{xy} if i and j have values $v_i(g) = x$ and $v_j(g) = y$ for good g, respectively; so there are four different types of goods: T_{aa}, T_{ab}, T_{ba} and T_{bb}. If $\min_{g \in A_j} v_i(g) = a$ or $\max_{g \in A_j} v_i(g) = b$, then i is EFX towards j since i is EF1 towards j, and the two notions coincide in this case for the pair (i, j); recall that any MNW allocation is EF1 [Caragiannis et al., 2019b] . Therefore, from now on, we will assume that $\min_{g \in A_j} v_i(g) = b$ and $\max_{g \in A_j} v_i(g) = a$, which implies that $|A_j| \geq 2$ and A_j includes at least one good of type T_{ba} or T_{bb}.

<u>Case I</u>: *There is at least one good of type* T_{bb} *in* A_j.

<u>*Subcase (a):*</u> A_j *does not include any good of type* T_{ab}. Assume, towards a contradiction, that i is not EFX towards j: $v_i(A_i) < v_i(A_j) - b$. Since $v_j(g) \geq v_i(g)$ for all $g \in A_j$, we have that $v_j(A_j) \geq v_i(A_j)$. We now define a new allocation by moving a good $h \in A_j$ of type T_{bb} from j to i. In this new allocation, the product of the values of i and j is

$$\begin{aligned}(v_i(A_i) + b)(v_j(A_j) - b) &= v_i(A_i)v_j(A_j) + b(v_j(A_j) - v_i(A_i) - b) \\ &\geq v_i(A_i)v_j(A_j) + b(v_i(A_j) - v_i(A_i) - b) \\ &> v_i(A_i)v_j(A_j).\end{aligned}$$

Since the allocation of all other agents has not been changed, the new allocation achieves a strictly larger Nash welfare than $\mathbf{A}^*$, yielding a contradiction.

<u>*Subcase (b):*</u> A_j *includes at least one good* g *of type* T_{ab}. We will argue about the structure of set A_i. If A_i includes any good x of type T_{aa}, T_{ba} or T_{bb}, then by exchanging g with x, we obtain an allocation with strictly higher Nash welfare, contradicting the choice of $\mathbf{A}^*$. For example, if x is of type T_{aa}, then in the new allocation (after swapping x and g) agent i has exactly the same value, but agent j's value has strictly increased by an amount $a - b > 0$. One can verify that the same holds for the other two types. Hence, A_i must include only goods of type T_{ab}, which implies that $v_i(A_i) = |A_i|a$.

Towards a contradiction, assume that i is not EFX towards j. If $|A_j| \leq |A_i| + 1$, since A_j includes some good h for which $v_i(h) = b$, we have that

$$v_i(A_j) \leq (|A_j| - 1)a + b \leq |A_i|a + b = v_i(A_i) + b,$$

i.e., agent i is EFX towards j. So, it must be $|A_j| \geq |A_i| + 2$. We create a new allocation by moving a good $g \in T_{ab}$ from j to i. The product of the values of i and j then becomes

$$(v_i(A_i) + a)(v_j(A_j) - b) = v_i(A_i)v_j(A_j) + av_j(A_j) - bv_i(A_i) - ab.$$

Since $v_j(A_j) \geq |A_j|b \geq (|A_i|+2)b$ and $v_i(A_i) = |A_i|a$, we have that

$$av_j(A_j) - bv_i(A_i) - ab \geq (|A_i|+2)ab - |A_i|ab - ab = ab > 0.$$

Since the bundles of the other agents have not been changed, we have that the new allocation has strictly larger Nash welfare than $\mathbf{A}^*$, contradicting its choice.

<u>Case II</u>: *There are no goods of type T_{bb} in A_j.*

Then A_j includes at least one good of type T_{ba}. If A_j includes at least one good of type T_{ab}, then, as we argued in Case I(b) above, in order for $\mathbf{A}^*$ to be an MNW allocation, A_i cannot include any goods of type T_{aa}, T_{ba} or T_{bb}. As a result, A_i includes only goods of type T_{ab} and by reproducing the analysis used in Case I(b) it follows that $\mathbf{A}^*$ is EFX.

So, we may assume that A_j includes goods of type T_{ba} and T_{aa} only. This implies that $v_j(A_j) = |A_j|a$. Assume towards a contradiction that i is not EFX towards j: $v_i(A_i) < v_i(A_j) - b$. Since A_j contains at least one good that i values as b, we also have that $v_i(A_j) \leq (|A_j|-1)a + b$. Combining the last two expressions, we obtain that

$$v_i(A_i) + a < |A_j|a = v_j(A_j).$$

Now, consider the allocation that is obtained from $\mathbf{A}^*$ by moving a good of type T_{aa} from j to i. We know that such a good exists since $\max_{g \in A_j} v_i(g) = a$. By using the last inequality, the product of the values of i and j in the new allocation is

$$\begin{aligned}(v_i(A_i)+a)(v_j(A_j)-a) &= v_i(A_i)v_j(A_j) + a(v_j(A_j) - v_i(A_i) - a)\\ &> v_i(A_i)v_j(A_j),\end{aligned}$$

which combined with the fact that the bundles of the other agents have not been changed, contradicts the choice of $\mathbf{A}^*$.

In any case, we conclude that $\mathbf{A}^*$ must be EFX. $\square$

Caragiannis et al. [2019a] presented a 3-value instance in which no MNW allocation is EFX. For completeness, we include here a simpler such instance, which further shows that the implication MNW $\Rightarrow$ $\{\text{EFX}, \text{EFX}_0\}$ is no longer true even for interval-value instances in which the length of the interval is almost zero. Let ε be a small positive constant and consider an instance with two agents and three goods with values as shown in the table:

	g_1	g_2	g_3
agent 1	$1-\varepsilon$	1	$1+\varepsilon$
agent 2	1	$1-\varepsilon$	$1+\varepsilon$

This is a 3-value instance with values $\{1-\varepsilon, 1, 1+\varepsilon\}$. Clearly, it is also an interval-value instance with interval of length 2ε, which can be arbitrarily close to zero by selecting ε to be extremely small. It is easy to verify that there are exactly two allocations achieving the maximum Nash welfare of $2+\varepsilon$: $\mathbf{A}_1 = (\{g_2\}, \{g_1, g_3\})$ and $\mathbf{A}_2 = (\{g_2, g_3\}, \{g_1\})$. The Nash welfare of any other allocation is either $2(1-\varepsilon)$ or $2+\varepsilon-\varepsilon^2$. Now, for $\ell \in \{1,2\}$, observe that in $\mathbf{A}_\ell$ agent ℓ is not EFX towards the other agent since she envies her even after the removal of g_ℓ.

3.2 On the complexity of maximizing the Nash welfare

We now turn our attention to the complexity of computing a maximum Nash welfare allocation. This problem is already known to be hard for many domain restrictions, and easy for only a few special cases. Nevertheless, its complexity for k-value instances with $k \in \{2,3,4\}$ has been open. Here we make significant progress towards settling these cases. We again start with the binary case.

Theorem 3.3. *For binary instances, computing an MNW allocation (and thus an* EFX_0 *allocation) can be done in polynomial time.*

Proof. Consider any binary instance $I = (N, M, (v_i)_{i\in N})$ and let $I_{>0}$ be the sub-instance consisting only of the goods that are positively valued by some agent. Given the MNW allocation $\mathbf{B}^*$ for $I_{>0}$, we can obtain the MNW allocation $\mathbf{A}^*$ for I by augmenting $\mathbf{B}^*$ so that all the zero-valued goods are given to the agent with minimum value according to $\mathbf{B}^*$. So, it remains to compute $\mathbf{B}^*$ in $I_{>0}$.

To do this, we use the greedy algorithm Alg-Binary of Barman et al. [2018b] which outputs an allocation maximizing the Nash welfare for the binary instance. Let $\mathbf{B}$ be the allocation that Alg-Binary outputs when given as input $I_{>0}$. If $\text{NW}(\mathbf{B}) > 0$, then $\mathbf{B}^* = \mathbf{B}$ is also EFX_0 by Theorem 3.1. However, if $\text{NW}(\mathbf{B}) = 0$, in which case all allocations have zero Nash welfare, $\mathbf{B}$ might *not* be an MNW allocation in our sense, i.e., an allocation that maximizes the number of agents with positive value and then the product of their values. Hence, $\mathbf{B}$ may not be EFX_0. To circumvent this, we define a bipartite graph consisting of nodes corresponding to the agents on the left and nodes corresponding to the goods on the right, while an edge between an agent and a good exists if the agent has value 1 for the good. By computing a maximum bipartite matching on this graph, it is guaranteed that the number of agents with positive value is maximized. Then, we run Alg-Binary on the restricted sub-instance of $I_{>0}$ where the set of agents includes only the ones that participate in the maximum matching, so that the product of their values (which now is going to be positive) is also maximized. This yields the desired allocation $\mathbf{B}^*$ with maximum Nash welfare for $I_{>0}$ in which the agents that did not participate in the maximum matching get an empty set. $\square$

For general 2-value instances we were unable to resolve the complexity of computing an MNW allocation, but we show that the problem is NP-complete for 3-value instances. This extends the hardness aspect (but not the inapproximability) of the result of Lee [2017] for 5-value instances.

Theorem 3.4. *Computing an MNW allocation is* NP*-complete, even for* 3*-value instances.*

Proof. We will prove that the problem of deciding whether there exists an allocation that achieves Nash welfare at least some value U is NP-complete. Given an allocation, it is trivial to check whether its Nash welfare is at least U. For the hardness, we give a reduction from a special version of 3SAT, known as 2P2N-3SAT, where every variable appears twice as a positive literal and twice as a negative literal. This problem is known to be NP-complete [Yoshinaka, 2005, Berman et al., 2003].

Consider an instance of 2P2N-3SAT, in which the set of variables is $\{x_1, \ldots, x_n\}$, and the set of clauses is $\{C_1, \ldots, C_m\}$. We will now describe how to construct a 3-value instance with $2m + 5n$ goods and $3m + 2n$ agents, where the set of values V consists of a, $b = 1$ and $c = 0$, where $a > 1/(\sqrt[2m]{2} - 1)$.

For each variable x_i, introduce two variable-agents $\{T_i, F_i\}$, as well as 5 variable-goods, denoted as $\{s_{i,0}, s_{i,1}, s_{i,2}, s_{i,3}, s_{i,4}\}$. The values of the variable-agents for the variable-goods are:

- $s_{i,0}$: both T_i and F_i have value a for it; the value of all other variable-agents is 0.
- $s_{i,1}$ and $s_{i,2}$: T_i has value 1 for each of them; the value of all other variable-agents is 0.
- $s_{i,3}$ and $s_{i,4}$: F_i has value 1 for each of them; the value of all other variable-agents is 0.

For each clause $C_j = (\ell_1 \vee \ell_2 \vee \ell_3)$, where ℓ_1, ℓ_2 and ℓ_3 are the three literals in the clause, introduce three clause-agents $\{C_j^1, C_j^2, C_j^3\}$ and two clause-goods $\{p_j, q_j\}$. The values of the clause-agents for all goods as well as the values of the variable-agents for the clause-goods are as follows:

- Both p_j and q_j are valued as a by the three corresponding clause-agents C_j^1, C_j^2 and C_j^3, and as 0 by all other (variable- and clause-) agents.
- For every j, the clause-agent C_j^t, $t \in [3]$ has value 0 for all other goods, besides one:
  - If $\ell_t = x_i$, then C_j^t has value 1 for one of two variable-goods $s_{i,1}, s_{i,2}$ (whichever is not valued by some other clause-agent).
  - If $\ell_t = \overline{x_i}$, then C_j^t has value 1 for one of the two variable-goods $s_{i,3}, s_{i,4}$ (whichever is not valued by some other clause-agent).

Note that since there are exactly two occurrences of x_i and $\overline{x_i}$ in the 2P2N-3SAT instance, each of the variable-goods $s_{i,1}, s_{i,2}, s_{i,3}, s_{i,4}$ is valued by exactly two agents (and one of them is always T_i or F_i).

Given a satisfying assignment for the 2P2N-3SAT instance, we define the following allocation with Nash welfare at least $U = 2^n a^{2m+n}$:

- Variables: If $x_i = 1$, then we allocate $s_{i,0}$ to T_i (for value a) and $\{s_{i,3}, s_{i,4}\}$ to F_i (for value 2). Otherwise ($x_i = 0$), we allocate $s_{i,0}$ to F_i (for value a) and $\{s_{i,1}, s_{i,2}\}$ to T_i (for value 2).

- Clauses: Let $C_j = (\ell_1 \vee \ell_2 \vee \ell_3)$. Since C_j is satisfied, at least one of ℓ_1, ℓ_2 or ℓ_3 is true; without loss of generality, assume that ℓ_1 is true. Then, we allocate p_j to C_j^2 (for value a), and q_j to C_j^3 (again for value a). Note that there is now exactly one good valued by C_j^1 that is still unallocated. If $\ell_1 = x_i$, then this good is one of $s_{i,1}$ or $s_{i,2}$, while if $\ell_1 = \overline{x_i}$, then this good is one of $s_{i,3}$ or $s_{i,4}$. In any case, we allocate this good to C_j^1.

- We allocate any remaining goods arbitrarily.

Now observe that for any variable x_i the product of the values of the two variable-agents T_i and F_i is at least $2a$, and for any clause $C_j = (\ell_1 \vee \ell_2 \vee \ell_3)$ the product of the values of the three clause-agents C_j^1, C_j^2 and C_j^3 is at least a^2. Consequently, the Nash welfare of the resulting allocation is at least $(2a)^n(a^2)^m = 2^n a^{2m+n}$.

Conversely, we will now show that given any allocation with Nash welfare at least $2^n a^{2m+n}$, we can obtain a satisfying assignment for the 2P2N-3SAT instance. Without loss of generality, we can assume that every good has been allocated to some agent that has positive value for it. Otherwise, we can modify the allocation so that this holds and the Nash welfare will not decrease. In particular, in any clause C_j, the goods p_j and q_j will be allocated to the agents C_j^1, C_j^2, C_j^3.

More specifically, we can assume that p_j and q_j have been allocated to distinct clause-agents. Suppose otherwise that p_j and q_j have both been allocated to the same agent, say C_j^1. If one of C_j^2 or C_j^3 has value 0 (and hence the Nash welfare is 0), then by allocating q_j to that agent instead the Nash welfare will not decrease. If both C_j^2 and C_j^3 have positive value, then this must be equal to 1. Reallocating q_j to (say) C_j^2 will not decrease the Nash welfare. Indeed, before the modification the product of values was at most $2a+1$ and after it is at least $a(a+1)$. By our choice of a, we have $2a+1 \leq a(a+1)$, so the Nash welfare cannot decrease as a result of this modification.

By the observation above, we can assume that $s_{i,0}$ has been allocated to T_i or F_i. If it has been allocated to T_i, then we can also assume that goods $s_{i,1}$ and $s_{i,2}$ have not been allocated to T_i, but instead that they have each been allocated to the unique other agent that values them. Let us show this for $s_{i,1}$ (the other case is identical). Assume that $s_{i,1}$ has been allocated to T_i and let C_j^t be the unique other agent that values it. By the above discussion, C_j^t has obtained at most one of p_j and q_j. If C_j^t has value 0, it is easy to see that allocating $s_{i,1}$ to C_j^t does not decrease the Nash welfare. If C_j^t has positive value, then this must be equal to a. Now if T_i has utility $a+2$, then by allocating $s_{i,1}$ to C_j^t instead, the product of values increases from $(a+2)a$ to $(a+1)^2$. On the other hand, if T_i has utility $a+1$, then by allocating $s_{i,1}$ to C_j^t instead, the product of values remains the same (it goes from $(a+1)a$ to $a(a+1)$). Similarly, if good $s_{i,0}$ has been allocated to F_i, we can assume that goods $s_{i,3}$ and $s_{i,4}$ have not been allocated to T_i, but instead that they have each been allocated to the unique other agent that values them.

We now construct an assignment for the variables as follows:

- If T_i has obtained $s_{i,0}$, then we set $x_i = 1$.

- If F_i has obtained $s_{i,0}$, then we set $x_i = 0$.

We say that the allocation satisfies the *consistency* property, if the agent in $\{T_i, F_i\}$ that has not obtained $s_{i,0}$, has obtained all other goods which she values for a total value of 2.

Let us first show that if the allocation satisfies the consistency property, the assignment satisfies the 2P2N-3SAT instance. Observe that for every clause C_j, one of the three agents C_j^1, C_j^2 or C_j^3 will not obtain one of the two clause-goods which they all value as a. Thus, this agent has to obtain the single other variable-good which she positively values, otherwise the Nash welfare would be 0. By the consistency property, it follows that the literal associated to this agent must be satisfied. Thus, the clause is satisfied.

Finally, let us show that if the Nash welfare of the allocation is at least $2^n a^{2m+n}$, then the consistency property holds. Assume that there exists an i such that the agent in $\{T_i, F_i\}$ that has not obtained $s_{i,0}$, has value strictly less than 2, i.e. at most 1. Thus, the product of the values of T_i and F_i is at most a. For any other i', the product of the values of $T_{i'}$ and $F_{i'}$ is at most $2a$. For any clause C_j, the product of the values of the three corresponding clause-agents is at most $(a+1)^2$. Thus, altogether the Nash welfare is at most $(a+1)^{2m}(2a)^{n-1}a = 2^{n-1}(a+1)^{2m}a^n$. By our choice of $a > 1/(\sqrt[2m]{2} - 1)$, we have ensured that $(a+1)^{2m} < 2a^{2m}$, and so the Nash welfare is indeed strictly less than $2^n a^{2m+n}$. □

4 Computing EFX Allocations for Restricted Domains

Even though we showed that any MNW allocation is also EFX_0 for agents with 2-value valuation functions, it remains an open question whether there exists a polynomial-time algorithm for computing such allocations beyond the binary case. In this section, we try to circumvent computing MNW allocations and aim to design efficient algorithms for computing EFX_0 allocations (which might not maximize the Nash welfare) for 2-value instances and interval-value instances.

4.1 2-value instances

We begin with considering 2-value instances with values $\{a, b\}$ such that $a > b \geq 0$. Our algorithm, which we call MATCH&FREEZE, proceeds in rounds and maintains a set of *active* agents L, initially containing everyone. In each round, every active agent is given exactly one of the remaining goods, with the possible exception of the last round in which there might not be enough goods left for all agents. The algorithm terminates when all goods have been allocated.

To determine which good each active agent gets during a round, we create a bipartite graph $G = (L \cup R, E)$ with nodes corresponding to the active agents L on one side and to the remaining goods R on the other. An edge between an active agent i and a good g exists if and only if $v_i(g) = a$. We first compute a maximum matching on this graph. Then each agent gets the good to which she is matched. If there are

Algorithm 1 Match&Freeze$(N, M, (v_i)_{i \in N})$

Input: a 2-value instance using the values a, b ($a > b \geq 0$)
$L \leftarrow N$ ▷ set of active agents
$R \leftarrow M$ ▷ set of unallocated goods
$\ell = (1, 2, \dots, n)$ ▷ ordered list of agents
while $R \neq \emptyset$ **do** ▷ every iteration is a round
  Construct the bipartite graph $G = (L \cup R, E)$.
  Compute a maximum matching on G.
  for each matched pair (i, g) **do**
    Allocate good g to agent i.
    Remove g from R.
  for each unmatched active agent i w.r.t. ℓ **do**
    Allocate *one* arbitrary unallocated good g to i.
    Remove g from R.
  Construct the set F of agents that need to freeze, using procedure Construct-Frozen.
  Remove agents of F from L for the next $\lfloor a/b - 1 \rfloor$ rounds.
  Put agents of F to the end of ℓ.
return the resulting allocation $\mathbf{A}$.

procedure Construct-Frozen: ▷ g_i denotes the good allocated to agent i in the current round
Construct $F := \{i \in L \mid \exists j \in L : v_j(g_i) = a, v_j(g_j) = b\}$.
while there exists $i \in L \setminus F$ such that $a \in \{v_j(g_i) \mid j \in F\}$ **do**
  Add i to F.
return the set of agents F that will freeze.

agents who are not matched to any good and there are still available goods, the unmatched agents receive one arbitrary available good each (subject to availability).

There are two possible reasons why an agent i is not matched to any good in a round: (1) she does not have value a for any good (only b), or (2) the maximum matching is such that all goods for which her value is a are given to other agents. Case (1) does not affect whether the final allocation will be EFX$_0$, but case (2) is crucial. This is because agent i might now have much smaller value for her own bundle compared to her value for the bundles of some agents that *just received* one good each that i values as a. Let Z be the set of these agents. To make up the distance, agent i should possibly receive multiple goods of value b while all agents in Z must *freeze* for a number of subsequent rounds depending on the ratio a/b.

We define the set F of agents that need to freeze at the end of round r to consist of all those agents who must become inactive because they have obtained too much value from the perspective of other agents (similarly to Case (2) above). Formally, for every active agent i, let g_i be the good she gets in round r. We begin by setting $F = \{i \in L \mid \exists j \in L : v_j(g_i) = a, v_j(g_j) = b\}$. Then, iteratively, as long as there is an agent $i \in L \setminus F$ such that there exists $j \in F$ with $v_j(g_i) = a$, we also add i to F. This construction is implemented by the procedure Construct-Frozen in the Match&Freeze algorithm. Each agent in F will remain *frozen* for the next $\lfloor a/b - 1 \rfloor$ rounds. In the case $b = 0$, we use the interpretation $\lfloor a/b - 1 \rfloor = +\infty$ in which case agents in F remain frozen forever. Exploiting the properties of the maximum matchings used to allocate goods we can prove that no agent in F will become envious while frozen, and that F is always a strict subset of L. The latter means that there is at least one (non-frozen) active agent at any time, and thus the algorithm will terminate after at most m rounds.

Theorem 4.1. *For any* 2*-value instance,* Match&Freeze *computes an* EFX_0 *allocation in polynomial time.*

Proof. Let $\mathbf{A} = (A_i)_{i \in N}$ be the allocation outputted by Match&Freeze when given as input a 2-value instance $I = (N, M, (v_i)_{i \in N})$ such that $v_i(g) \in \{a, b\}$, $a > b \geq 0$, for every $i \in N$ and $g \in M$. For any agent i, let r_i be the round in which the last goods of value a for agent i were allocated. We will need the following two lemmas.

Lemma 4.2. *For every agent* i*, it holds that:*

- *She was allocated a good for which she has value* a *in each of the rounds* $1, 2, \ldots, r_i - 1$.
- *She can freeze only at the end of round* r_i*, and only if during that round she got a good for which she has value* a.
- *She can freeze at most once. After freezing, she has value* b *for each of the remaining goods.*

Proof of Lemma 4.2. Note that the third part of the lemma immediately follows from the second part and the definition of r_i. We begin by proving the second part of the lemma. Consider an agent i that got frozen at the end of some round r. For any agent j that was allocated a good in round r, let g_j denote that good. Since agent $i = i_0$ got frozen, by construction of the set of frozen agents (see Construct-Frozen), there exist distinct agents $i_1, \ldots, i_k$ ($k \leq n$) such that $v_{i_\ell}(g_{i_{\ell-1}}) = a$ for $\ell \in [k]$, and $v_{i_k}(g_{i_k}) = b$. Note, in particular, that agent i_k was not matched in round r, and that agent $i = i_0$ must have obtained a good in round r. Let us now show that the following cases are impossible:

- $v_i(g_i) = b$. This means that agent $i = i_0$ was also not matched in round r, and thus out of the agents $i_0, \ldots, i_k$ at most $k - 1$ were matched in round r. However, since $v_{i_\ell}(g_{i_{\ell-1}}) = a$ for every $\ell \in [k]$, we could match agent i_ℓ to good $g_{i_{\ell-1}}$, and ensure that at least k out of the agents $i_0, \ldots, i_k$ are matched, without changing the matching for any other agents. This is a contradiction, since the algorithm uses a maximum matching. Therefore, it must be $v_i(g_i) = a$.

- $r > r_i$. By the definition of r_i, agent i must have been allocated a good she values as b in round r, which is impossible as we showed above. Hence, it must be $r \leq r_i$.
- $r < r_i$. By the definition of r_i, at the end of round r there exists an unallocated good g^* such that $v_i(g^*) = a$. Since agent i_k was not matched in round r, this means that at most k of the agents $i_0, \ldots, i_k$ were matched in that round. However, by matching agent i_ℓ to good $g_{i_{\ell-1}}$ for each $\ell \in [k]$, and $i = i_0$ to g^*, we can ensure that all of these $k + 1$ agents are matched, without changing the matching for any other agents. This is again a contradiction, since the algorithm uses a maximum matching. Therefore, it must be $r \geq r_i$.

By the last two cases, we have that agent i can only get frozen in round $r = r_i$, and by the first case, this can only happen if agent i obtains a good she values a in that round.

Finally, we prove the first part of the lemma. Note, first of all, that by the second part proved above, agent i will not be frozen in any of the rounds $1, 2, \ldots, r_i - 1$, and will thus be allocated a good in each of those rounds. Assume that agent i was allocated a good she values as b in some round $r < r_i$. This would mean that agent i was left unmatched in round r. However, in round $r_i > r$ there were still goods that i values as a available. Consequently, agent i could also have been matched in round r, which means that we did not use a maximum matching, a contradiction. □

Lemma 4.3. *If at the beginning of some round $r > r_i$, an active agent i is envy-free towards agent j, then agent i will be* EFX_0 *towards agent j at the end of the algorithm.*

Proof of Lemma 4.3. By the second part of Lemma 4.2, agent i is active during all rounds after $r > r_i$ until the end of the algorithm, and i has value b for all remaining goods. Consequently, agent i will be allocated a good she values as b in each subsequent round, except potentially the last round (during which she may not get any good), while the value of i for the bundle of agent j can increase by at most b in each subsequent round as well. Consequently, when the algorithm terminates, $v_i(A_j)$ can be at most b more than $v_i(A_i)$. □

It is easy to see that the algorithm terminates in polynomial time. If no agent ever gets frozen, then the algorithm terminates after at most $\lceil m/n \rceil$ rounds. Otherwise, let r be the first round in which an agent gets frozen. For any agent j that was allocated a good in round r, let g_j denote that good. Since an agent was frozen in round r, there exists some agent j with $v_j(g_j) = b$ and some agent i with $v_j(g_i) = a$. By the definition of r_j, it follows that $r \leq r_j$, since good g_i, which was allocated in round r, has value a for agent j. By the first statement in Lemma 4.2, it follows that $r \geq r_j$, since agent j was allocated a good for which she has value b in round r. As a result, we have that $r = r_j$. Furthermore, by the second statement in Lemma 4.2, we know that agent j did not get frozen at the end of round $r = r_j$ (because $v_j(g_j) = b$), which means that she never gets frozen. Since j gets a good in every round, the algorithm terminates after at most m rounds.

Let us now show that the algorithm constructs an EFX_0 allocation. If some agent i has value b for all goods, then by the argument in the proof of Lemma 4.3, agent i will be EFX_0 towards all other agents at the end of the algorithm. If there is at least one good which i values as a, then r_i is well-defined. Recall that by Lemma 4.2, it holds that agent i is allocated a good she values as a in each of the rounds $1, 2, \ldots, r_i - 1$. In round r_i there are two cases:

<u>Case I</u>: *Agent i is allocated a good she values as a.* Then, at the end of round r_i, agent i has total value ar_i for her bundle and total value at most ar_i for the bundle of any other agent. If agent i did not become frozen at the end of round r_i (which means she never will), then she will be EFX_0 towards all agents at the end of the algorithm by Lemma 4.3. Now consider the case where agent i froze at the end of round r_i.

- Any agent j who obtained a good that i values as a in round r_i will also freeze at the end of round r_i (by the definition of the set F of frozen agents), and both i and j will re-enter in the same round later. Thus, since i is envy-free towards j up until round r_i and both agents will never freeze again, i will be EFX_0 towards j at the end of the algorithm by Lemma 4.3.

- For any agent j who obtained a good that i values as b in round r_i, the value of i for the bundle of j at the end of round r_i is at most $a(r_i - 1) + b$. Hence, when agent i re-enters (after the end of round $r_i + \lfloor a/b - 1 \rfloor$), her value for j's bundle can be at most $a(r_i - 1) + b + b\lfloor a/b - 1 \rfloor \leq ar_i$. Thus, agent i will be envy-free towards agent j at this point, and EFX_0 towards j at the end of the algorithm by Lemma 4.3. If the algorithm terminates before agent i re-enters (which could happen if $b = 0$), then agent i is envy-free towards j.

<u>Case II</u>: *Agent i is allocated a good she values as b.* At the end of round r_i her total value is $a(r_i - 1) + b$, and she will never freeze by Lemma 4.2.

- For any agent j who obtained a good that i values as b in round r_i, the value of i for j's bundle can be at most $a(r_i - 1) + b$. Thus, agent i is envy-free towards j at the end of r_i, and EFX_0 at the end of the algorithm by Lemma 4.3.

- Any agent j who obtained a good that i values as a in round r_i must have frozen at the end of round r_i, by the construction of the set of frozen agents. If the algorithm terminates before agent j re-enters (for example, if $b = 0$), then agent i is EFX_0 towards j, because the value of i for j's bundle is at most ar_i and the least valuable good from i's perspective is of value a. Otherwise, agent j re-enters before the termination of the algorithm after the end of round $r_i + \lfloor a/b - 1 \rfloor$, and the value of i for j's bundle is still at most ar_i, as j did not receive any other good. Since i's own value increased to $a(r_i - 1) + b + b\lfloor a/b - 1 \rfloor > ar_i - b$, her envy towards agent j is at most b at this point. Furthermore, i has value b for the remaining goods. Note that when agents are frozen, the algorithm moves them to the end of the list ℓ (line 16). As a result, in the last round of the algorithm, when we allocate the remaining goods to unmatched agents (line 11), agents that have never gotten frozen are prioritized over agents who were frozen at some point in the algorithm. In particular, it holds that in any round, it is not possible that agent j gets a good, but not agent i. By an argument similar to the one used in the proof of Lemma 4.3, it follows that the envy will still be at most b at the end of the algorithm.

This completes the proof. □

4.2 Interval-value instances

From our discussion thus far, it may seem like the difficultly of proving the existence of EFX_0 allocations is directly related to the number of different values that the agents have, but this is not entirely true. We will now show that the range between the lowest and the highest value also plays a very important role: for specific ranges, and independently of the number of values therein (which can be infinite), computing EFX allocations can be achieved by very simple algorithms. In particular, we show that EFX allocations exist for interval-instances in which the values of each agent i are in some interval $[x_i, 2x_i]$, $x_i \in \mathbb{R}_{>0}$, by using a simple modification of the *round-robin algorithm*.

According to this algorithm, we fix an ordering of the agents and then they simply pick their favorite unallocated good one by one, with respect to that ordering. This continues in rounds of n goods each, until we reach a point where there are not enough goods for everyone. For this last round (if it exists), the agents pick in reverse order (see Algorithm 2).

Theorem 4.4. *Given an interval-instance in which the values of agent i are in the interval $[x_i, 2x_i]$, $x_i \in \mathbb{R}_{>0}$,* Modified round-robin *computes an EFX allocation in polynomial time.*

Proof. Let $\mathbf{A} = (A_1, \dots, A_n)$ be the allocation produced by Algorithm 2. First observe that if $m < n$ then the statement holds trivially. Hence, we assume that $m = kn + \ell$ for some $k \geq 1$ and $0 \leq \ell < n$. In this case, the first $n - \ell$ agents will get k goods and the last ℓ agents will get $k + 1$ goods. Consider now an agent i and let g_{ir} be the good that she gets in round $1 \leq r \leq k+1$. We will show that i is EFX towards any other agent j, by distinguishing between cases depending on whether i selects before or after j according to the main ordering of the algorithm.

Case I: $i < j$. Agent j either has the same number of goods as i, or one more good than i.

- If both agents get k goods, since i always chooses her most-valuable good before j, we have that $v_i(g_{ir}) \geq v_i(g_{jr})$ for every $r \in [k]$, and thus i does not envy j.

Algorithm 2 Modified round-robin

```
Input: Instance with m = kn + ℓ, k ≥ 0, 0 ≤ ℓ < n
for i = 1, ..., n do
    A_i = ∅
S = M
for r = 1, ..., k do
    for i = 1, ..., n do
        g ∈ arg max_{q∈S} v_i(q)
        A_i = A_i ∪ {g}
        S = S \ {g}
for i = n, ..., n − ℓ + 1 do
    g ∈ arg max_{q∈S} v_i(q)
    A_i = A_i ∪ {g}
    S = S \ {g}
return A = (A_1, ..., A_n)
```

- If both agents get $k+1$ goods or agent i has k goods and agent j has $k+1$ goods, let $g_{jt} \in A_j$ be the least-valuable good according to agent i, which j gets during round $t \in [k+1]$. If $t = k+1$, then since $v_i(g_{ir}) \geq v_i(g_{jr})$ for every $r \in [k]$, we immediately obtain that $v_i(A_i) \geq v_i(A_j \setminus \{g_{j,t}\})$. If $t \leq k$, since $v_i(g_{it}) \geq v_i(g_{j,k+1})$ and $v_i(g_{ir}) \geq v_i(g_{jr})$ for every $r \in [k] \setminus \{t\}$, we again obtain that $v_i(A_i) \geq v_i(A_j \setminus \{g_{jt}\})$.

Case II: $i > j$. Agent j has either the same number of goods as i or one less good than i. Once again, let $g_{jt} \in A_j$ be the least-valuable good according to agent i, where t is the round during which agent j obtained the good. Now observe that in general we have that $v_i(g_{ir}) \geq v_i(g_{j,r+1})$ for every $r \in [k-1]$. If $t = 1$, then the statement holds trivially. So, we assume that $t \geq 2$.

- If agent j gets k goods, then we have that $v_i(g_{ir}) \geq v_i(g_{j,r+1})$ for every $1 \leq r \leq t-2$ and $v_i(g_{ir}) \geq v_i(g_{j,r+2})$ for every $t-1 \leq r \leq k-2$. So far we have bounded all goods in $A_j \setminus \{g_{jt}\}$ besides g_{j1}, using all goods in A_i besides $g_{i,k-1}$ and $g_{i,k}$ (and also $g_{i,k+1}$ if it exists). Since all the values lie in the interval $[x_i, 2x_i]$, we have that $v_i(g_{i,k-1}) + v_i(g_{ik}) \geq v_i(g_{j1})$, which yields that $v_i(A_i) \geq v_i(A_j \setminus \{g_{jt}\})$.
- If both agents get $k+1$ goods, then we have that $v_i(g_{ir}) \geq v_i(g_{j,r+1})$ for every $1 \leq r \leq t-2$ and $v_i(g_{ir}) \geq v_i(g_{j,r+2})$ for every $t-1 \leq r \leq k-1$. Again, the values of the goods in $A_j \setminus \{g_i t\}$ besides g_{j1} have been bounded by the values of the goods in A_i besides g_{ik} and $g_{i,k+1}$. Since all the values lie

in the interval $[x_i, 2x_i]$, we have that $v_i(g_{ik})+v_i(g_{i,k+1}) \geq v_i(g_{j1})$, and thus $v_i(A_i) \geq v_i(A_j \setminus \{g_{jt}\})$.

This completes the proof. $\square$

5 MNW and the EFX-value

As we saw in Section 3, maximizing the Nash welfare does not yield an EFX allocation in general. Here we take a different route and instead of considering exact EFX allocations, we focus on approximation. We start by showcasing that maximizing the Nash welfare does not guarantee *any* meaningful approximation of EFX according to the current definition of approximation used in the literature [Plaut and Roughgarden, 2018, Amanatidis et al., 2018, 2020, Chan et al., 2019].

Definition 5.1 (α-EFX allocation)**.** For $\alpha \in (0, 1]$, an allocation $\mathbf{A}$ is α-EFX if for every pair $i, j \in N$ and every good $g \in A_j$ such that $v_i(g) > 0$, it holds that $v_i(A_i) \geq \alpha v_i(A_j \setminus \{g\})$.

Let $w > 1$ and $\varepsilon < \frac{1}{2w}$. Consider the following very simple instance with two agents, three goods, and values given in the table:

	g_1	g_2	g_3
agent 1	w	0	1/2
agent 2	w	1	ε

We first claim that the allocation $\mathbf{A}^* = (A_1 = \{g_1, g_3\}, A_2 = \{g_2\})$ is the only one that achieves the maximum Nash welfare of $w + 1/2$. Indeed, the Nash welfare of any allocation that gives g_2 to agent 1 can only increase by moving g_2 to agent 2, while any allocation other than $\mathbf{A}^*$ that gives g_2 to agent 2 has Nash welfare either $(w + 1)/2$ or $w + \varepsilon w < w + 1/2$. Notice, however, that $\mathbf{A}^*$ is not EFX since $v_2(A_2) = 1 < w = v_2(A_1 \setminus \{g_3\})$. Instead, it is only $1/w$-EFX, an approximation factor that can be arbitrarily close to zero as w becomes large.

Nevertheless, $\mathbf{A}^*$ is not that far away from being an EFX allocation! To see this, consider the allocation $\mathbf{B} = (B_1 = \{g_1\}, B_2 = \{g_2, g_3\})$ that is obtained from $\mathbf{A}^*$ by only moving g_3 from agent 1 to agent 2. Clearly, agent 2 is EFX towards agent 1, as the latter gets only one good. Moreover, the value agent 2 has now is $v_2(B_2) = 1 + \varepsilon$, which is extremely close to the value $v_2(A_2) = 1$ that agent 2 has in $\mathbf{A}^*$. So, even though $\mathbf{A}^*$ is $1/w$-EFX because $v_2(A_2)$ is very low compared to $v_2(A_1 \setminus \{g_3\})$, $v_2(A_2)$ is actually very close to the value she would have in a *nearby* EFX allocation. Consequently, if we accept that agent 2 considers the EFX allocation $\mathbf{B}$ as fair, then she should also consider $\mathbf{A}^*$ as being *almost* fair.

We say that the value $v_2(B_2) = 1 + \varepsilon$ is the EFX-*value* that agent 2 can achieve by augmenting her bundle with a subset of goods from agent 1 in order to create the closest-to-$\mathbf{A}^*$ (in terms of value) allocation $\mathbf{B}$ which she considers as EFX. Then, since $v_2(A_2) = \frac{1}{1+\varepsilon} v_2(B_2)$, agent 2 achieves an approximation of $\frac{1}{1+\varepsilon}$ of her EFX-value. Let us now formalize these notions for any number of agents.

Definition 5.2 (EFX-value)**.** Let $\mathbf{A} = (A_1, \ldots, A_n)$ be an allocation. For every pair of agents $i, j \in N$, let $X_{ij} \subseteq A_j$ be a set of goods such that $v_i(A_i \cup X_{ij}) \geq v_i(A_j \setminus (X_{ij} \cup \{g\}))$ for every $g \in A_j \setminus X_{ij}$ and $v_i(A_i \cup X_{ij})$ is minimized. Then, the EFX-*value* of agent i is

$$\boldsymbol{\chi}_i(\mathbf{A}) = \max_{j \in N \setminus \{i\}} v_i(A_i \cup X_{ij}).$$

Definition 5.3 (α-vEFX allocation)**.** For $\alpha \in (0, 1]$, an allocation $\mathbf{A}$ is α-vEFX if $v_i(A_i) \geq \alpha \boldsymbol{\chi}_i(\mathbf{A})$ for every $i \in N$.

We remark that using EFX_0 instead of EFX in the above definitions does not make any difference since adding zeros does not affect the EFX-value. Furthermore, observe that a 1-vEFX allocation is an EFX allocation but not necessarily an EFX_0 allocation.

Our first technical result in this section illustrates the connection between approximate EFX and vEFX allocations.

Theorem 5.4. *For $\alpha \in (0, 1)$, an α-*EFX *allocation is also an $\frac{\alpha}{1+\alpha}$-*vEFX *allocation and this guarantee is tight. On the other hand, an α-*vEFX *allocation is not guaranteed to be β-*EFX*, for any $\alpha, \beta \in (0, 1)$.*

Proof. Let $\mathbf{A}$ be an α-EFX allocation such that $v_i(A_i) \geq \alpha \cdot v_i(A_j \setminus \{g\})$, where $v_i(g) \in \min_{q \in A_j} v_i(q)$, for every pair of agents $i, j \in N$. We can equivalently write this inequality as $(1 + \alpha)v_i(A_i) \geq \alpha(v_i(A_i) + v_i(A_j \setminus \{g\})$, and since $\boldsymbol{\chi}_i(\mathbf{A}) \leq v_i(A_i \cup A_j \setminus \{g\})$, we obtain that

$$\frac{v_i(A_i)}{\boldsymbol{\chi}_i(\mathbf{A})} \geq \frac{v_i(A_i)}{v_i(A_i \cup A_j \setminus \{g\})} \geq \frac{\alpha}{1 + \alpha}.$$

Furthermore, by moving all goods in $A_j \setminus \{g\}$ from j to i, we obtain an allocation in which j gets only one good, and consequently i is EFX towards j. Hence, $\mathbf{A}$ is indeed $\frac{\alpha}{\alpha+1}$-vEFX.

For the upper bound, let $\alpha \in (0, 1)$ and consider an instance in which some agent i has values given by the following table:

	g_1	g_2	g_3
agent i	1	$1/\alpha$	$1/\alpha$

In the allocation $\mathbf{A}$ according to which $A_i = \{g_1\}$ and $A_j = \{g_2, g_3\}$ for some agent $j \neq i$, agent i is not EFX towards agent j. Since $v_i(A_i) = 1$ and $v_i(A_j \setminus \{g_2\}) = v_i(A_j \setminus \{g_3\}) = 1/\alpha$, we have that $\mathbf{A}$ is α-EFX. Now, by moving g_2 from j to i, i becomes EFX towards j, and achieves a total value of $v_i(A_i) + v_i(g_2) = 1 + 1/\alpha$, yielding an approximation of $\frac{1}{1+1/\alpha} = \frac{\alpha}{1+\alpha}$ on her EFX-value.

For the reverse relation, let $\alpha \in (0, 1)$ and $\gamma > \frac{1-\alpha}{\alpha}$. Consider an instance in which the values of some agent i are:

	g_1	g_2	g_3
agent i	1	γ	$(1-\alpha)/\alpha$

Let $\mathbf{A}$ be the allocation according to which $A_i = \{g_1\}$ and $A_j = \{g_2, g_3\}$ for some agent $j \neq i$. Since $v_i(A_i) = 1$, agent i can become EFX towards j, by acquiring g_3, in which case she achieves a value of $1/\alpha$. Hence, $\mathbf{A}$ is α-vEFX. However, since $v_i(g_2) = \gamma > \frac{1-\alpha}{\alpha} = v_i(g_3)$ and $v_i(A_j \setminus \{g_3\}) = \gamma$, $\mathbf{A}$ is only $1/\gamma$-EFX. This approximation can be arbitrarily close to 0 as γ becomes large. □

Even though maximizing the Nash welfare may not yield a β-EFX for any $\beta \in (0, 1)$, as follows from the discussion preceding Definition 5.2, it is guaranteed to produce a constant vEFX allocation.

Theorem 5.5. *Any maximum Nash welfare allocation* $\mathbf{A}^*$ *is* $1/2$-vEFX*, and this bound is tight.*

By using arguments similar to those in the proof of Theorem 5.4, we can show that any EF1 allocation is $1/2$-vEFX. Then, Theorem 5.5 follows from the result of Caragiannis et al. [2019b] about MNW implying EF1. Below, we present a direct and self-contained proof of Theorem 5.5.

Proof. Consider a pair of agents i and j. Let $A_i = \{a_1, ..., a_{|A_i|}\}$ and $A_j = \{b_1, ..., b_{|A_j|}\}$ be the sets of goods allocated to i and j according to $\mathbf{A}^*$, respectively. Without loss of generality, we may assume that $0 \leq v_i(b_1) \leq ... \leq v_i(b_{|A_j|})$. Let $S = \{b_1, ..., b_\lambda\}$, $\lambda < |A_j|$ be the subset of least-valued goods of A_j such that i is EFX towards j when given the set S (on top of A_i), but is not EFX towards j when given only the set $S \setminus \{b_\lambda\}$. We are going to show that $v_i(A_i) \geq v_i(S) \Leftrightarrow 2v_i(A_i) \geq v_i(A_i \cup S)$. If this is true, then since $\boldsymbol{\chi}_i(\mathbf{A}) \leq v_i(A_i \cup S)$, $\mathbf{A}^*$ must be $1/2$-vEFX.

Assume towards a contradiction that $v_i(A_i) < v_i(S)$. By its definition, S is such that

$$v_i(A_i) + v_i(S \setminus \{b_\lambda\}) < v_i(A_j \setminus (S \setminus \{b_\lambda\})) - v_i(b_\lambda)$$

or, equivalently,

$$v_i(A_i) + v_i(S \setminus \{b_\lambda\}) < v_i(A_j \setminus S).$$

Also $v_i(S \setminus \{b_\lambda\}) \geq 0$ implies that $v_i(A_i) < v_i(A_j \setminus S)$. Since $v_j(S) + v_j(A_j \setminus S) = v_j(A_j)$, it must be the case that one of $v_j(S)$ and $v_j(A_j \setminus S)$ is at least $\frac{v_j(A_j)}{2}$, while the other is at most this much.

- If $v_j(S) \leq \frac{v_j(A_j)}{2}$ and $v_j(A_j \setminus S) \geq \frac{v_j(A_j)}{2}$, then we define a new allocation in which all goods in S are moved from agent j to agent i. By our assumption that $v_i(A_i) < v_i(S)$, the product of the new values of the two agents is

$$\big(v_i(A_i) + v_i(S)\big) \cdot v_j(A_j \setminus S) > 2v_i(A_i) \cdot \frac{v_j(A_j)}{2} = v_i(A_i) \cdot v_j(A_j).$$

  Since the sets of goods given to the other agents have not been altered, the new allocation has strictly more Nash welfare than $\mathbf{A}^*$, a contradiction.

- If $v_j(S) \geq \frac{v_j(A_j)}{2}$ and $v_j(A_j \setminus S) \leq \frac{v_j(A_j)}{2}$, then we define another allocation in which all goods in $A_j \setminus S$ are moved from j to i. Since $v_i(A_i) < v_i(A_j \setminus S)$, the product of the new values of the two agents becomes

$$\big(v_i(A_i) + v_i(A_j \setminus S)\big) \cdot v_j(S) > 2v_i(A_i) \cdot \frac{v_j(A_j)}{2} = v_i(A_i) \cdot v_j(A_j),$$

again yielding a contradiction.

Consequently, it must be $v_i(A_i) \geq v_i(S)$, meaning that $\mathbf{A}^*$ is $1/2$-vEFX.

For the upper bound, consider again the instance used in Section 3 to show that an MNW allocation may not be EFX for 3-value instances. For ease of reference, we repeat the table containing the values of the two agents for the three goods here:

	g_1	g_2	g_3
agent 1	$1-\varepsilon$	1	$1+\varepsilon$
agent 2	1	$1-\varepsilon$	$1+\varepsilon$

As we argued in Section 3, the allocations $\mathbf{A}_1 = (\{g_2, g_3\}, \{g_1\})$ and $\mathbf{A}_2 = (\{g_2\}, \{g_1, g_3\})$ are the only Nash welfare maximizing ones. Both of these are $\frac{1}{2-\varepsilon}$-vEFX since the envious agent can get the least-valued good from the other agent (worth $1-\varepsilon$) and become EFX. □

6 Directions for Future Work

We studied the connection between two celebrated notions, that of maximum Nash welfare and envy-freeness up to any good. We showed that a maximum Nash welfare allocation is always EFX_0 for 2-value instances, while this implication is no longer true for k-value instances with $k \geq 3$. The first question that our work leaves open is whether it is possible to compute in polynomial-time an allocation that maximizes the Nash welfare for 2-value instances.

Nevertheless, for 2-value instances we presented a polynomial-time algorithm for computing an EFX_0 allocation. Due to its novelty, we believe that the idea of repeatedly computing maximum matchings and freezing certain agents whenever they acquire too much value (compared to other agents), might be a stepping stone for proving the existence of EFX_0 more generally. That being said, while generalizing our algorithm to k-value instances with $k \geq 3$ definitely deserves further investigation, it does seem to be a highly non-trivial task.

Finally, going beyond exact MNW or EFX allocations, we discussed the connection between MNW and approximate EFX allocations. While an MNW allocation does not necessarily provide any meaningful guarantee according to the commonly used definition of approximation, we showed that it does under a new interpretation of approximation via the EFX-value. A natural question is whether one can design polynomial-time algorithms with strong approximation guarantees for both the EFX-value and the Nash welfare.

Acknowledgement

This work has been partially supported by the ERC Advanced Grant 788893 AMDROMA "Algorithmic and Mechanism Design Research in Online Markets", the MIUR PRIN project ALGADIMAR "Algorithms, Games, and Digital Markets", the NWO Veni project VI.Veni.192.153, the ERC Starting Grant 639945 ACCORD "Algorithms for Complex Collective Decisions on Structured Domains", and an EPSRC doctoral studentship (Reference 1892947). We would like to thank Ioannis Caragiannis for fruitful discussions at early stages of this work.

References

Martin Aleksandrov and Toby Walsh. Greedy algorithms for fair division of mixed manna. *CoRR*, abs/1911.11005, 2019.

Georgios Amanatidis, Evangelos Markakis, Afshin Nikzad, and Amin Saberi. Approximation algorithms for computing maximin share allocations. *ACM Trans. Algorithms*, 13(4):52:1–52:28, 2017.

Georgios Amanatidis, Georgios Birmpas, and Vangelis Markakis. Comparing approximate relaxations of envy-freeness. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 42–48, 2018.

Georgios Amanatidis, Evangelos Markakis, and Apostolos Ntokos. Multiple birds with one stone: Beating 1/2 for EFX and GMMS via envy cycle elimination. *Theoretical Computer Science*, 841:94–109, 2020.

Moshe Babaioff, Tomer Ezra, and Uriel Feige. Fair and truthful mechanisms for dichotomous valuations. *CoRR*, abs/2002.10704, 2020.

Siddharth Barman and Sanath Kumar Krishna Murthy. Approximation algorithms for maximin fair division. In *Proceedings of the 2017 ACM Conference on Economics and Computation (EC)*, pages 647–664, 2017.

Siddharth Barman, Arpita Biswas, Sanath Kumar Krishna Murthy, and Yadati Narahari. Groupwise maximin fair allocation of indivisible goods. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, pages 917–924, 2018a.

Siddharth Barman, Sanath Kumar Krishnamurthy, and Rohit Vaish. Greedy algorithms for maximizing Nash social welfare. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 7–13, 2018b.

Siddharth Barman, Sanath Kumar Krishnamurthy, and Rohit Vaish. Finding fair and efficient allocations. In *Proceedings of the 2018 ACM Conference on Economics and Computation (EC)*, pages 557–574, 2018c.

Xiaohui Bei, Guangda Huzhang, and Warut Suksompong. Truthful fair division without free disposal. *Social Choice and Welfare*, 55(3):523–545, 2020.

Piotr Berman, Marek Karpinski, and Alex D. Scott. Approximation hardness of short symmetric instances of MAX-3SAT. *Electronic Colloquium on Computational Complexity (ECCC)*, (049), 2003.

Sylvain Bouveret, Yann Chevaleyre, and Nicolas Maudet. Fair allocation of indivisible goods. In *Handbook of Computational Social Choice*, pages 284–310. Cambridge University Press, 2016.

Eric Budish. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy*, 119(6):1061–1103, 2011.

Ioannis Caragiannis, Nick Gravin, and Xin Huang. Envy-freeness up to any item with high Nash welfare: The virtue of donating items. In *Proceedings of the 2019 ACM Conference on Economics and Computation (EC)*, pages 527–545, 2019a.

Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D. Procaccia, Nisarg Shah, and Junxing Wang. The unreasonable fairness of maximum Nash welfare. *ACM Trans. Economics and Comput.*, 7(3):12:1–12:32, 2019b.

Hau Chan, Jing Chen, Bo Li, and Xiaowei Wu. Maximin-aware allocations of indivisible goods. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 137–143, 2019.

Bhaskar Ray Chaudhury, Jugal Garg, and Kurt Mehlhorn. EFX exists for three agents. In *Proceedings of the 21st ACM Conference on Economics and Computation (EC)*, pages 1–19, 2020a.

Bhaskar Ray Chaudhury, Telikepalli Kavitha, Kurt Mehlhorn, and Alkmini Sgouritsa. A little charity guarantees almost envy-freeness. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2658–2672, 2020b.

Richard Cole and Vasilis Gkatzelis. Approximating the nash social welfare with indivisible items. *SIAM J. Comput.*, 47(3):1211–1236, 2018.

Richard Cole, Nikhil R. Devanur, Vasilis Gkatzelis, Kamal Jain, Tung Mai, Vijay V. Vazirani, and Sadra Yazdanbod. Convex program duality, Fisher markets, and Nash social welfare. In *Proceedings of the 2017 ACM Conference on Economics and Computation (EC)*, pages 459–460, 2017.

Duncan K. Foley. Resource allocation and the public sector. *Yale Economics Essays*, 7:45–98, 1967.

George Gamow and Marvin Stern. *Puzzle-Math.* Viking press, 1958.

Jugal Garg and Peter McGlaughlin. Improving Nash social welfare approximations. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence, (IJCAI)*, pages 294–300, 2019.

Jugal Garg and Setareh Taki. An improved approximation algorithm for maximin shares. In *Proceedings of the 21st ACM Conference on Economics and Computation (EC)*, pages 379–380, 2020.

Jugal Garg, Peter McGlaughlin, and Setareh Taki. Approximating maximin share allocations. In *Proceedings of the 2nd Symposium on Simplicity in Algorithms, SOSA@SODA*, pages 20:1–20:11, 2019.

Jugal Garg, Pooja Kulkarni, and Rucha Kulkarni. Approximating Nash social welfare under submodular valuations through (un)matchings. In *Proceedings of the Thirty-First Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2020. To appear.

Mohammad Ghodsi, Mohammad Taghi Hajiaghayi, Masoud Seddighin, Saeed Seddighin, and Hadi Yami. Fair allocation of indivisible goods: Improvements and generalizations. In *Proceedings of the 2018 ACM Conference on Economics and Computation (EC)*, pages 539–556, 2018.

Laurent Gourvès, Jérôme Monnot, and Lydia Tlilane. Near fairness in matroids. In *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI)*, volume 263, pages 393–398, 2014.

David Kurokawa, Ariel D. Procaccia, and Junxing Wang. Fair enough: Guaranteeing approximate maximin shares. *J. ACM*, 65(2):8:1–8:27, 2018.

Maria Kyropoulou, Warut Suksompong, and Alexandros A. Voudouris. Almost envy-freeness in group resource allocation. *Theoretical Computer Science*, 841:110–123, 2020.

Euiwoong Lee. APX-hardness of maximizing Nash social welfare with indivisible items. *Information Processing Letters*, 122:17–20, 2017.

Richard J. Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. On approximately fair allocations of indivisible goods. In *Proceedings 5th ACM Conference on Electronic Commerce (EC)*, pages 125–131, 2004.

Pasin Manurangsi and Warut Suksompong. Closing gaps in asymptotic fair division. *CoRR*, abs/2004.05563, 2020.

Evangelos Markakis. Approximation algorithms and hardness results for fair division with indivisible goods. In *Trends in Computational Social Choice*, chapter 12. AI Access, 2017.

Benjamin Plaut and Tim Roughgarden. Almost envy-freeness with general valuations. In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2584–2603, 2018.

Ariel D. Procaccia. Cake cutting algorithms. In *Handbook of Computational Social Choice*, pages 311–330. Cambridge University Press, 2016.

Hugo Steinhaus. The problem of fair division. *Econometrica*, 16:101–104, 1948.

Warut Suksompong. Approximate maximin shares for groups of agents. *Mathematical Social Sciences*, 92: 40–47, 2018.

Warut Suksompong. On the number of almost envy-free allocations. *Discrete Applied Mathematics*, 284: 606–610, 2020.

Hal R. Varian. Equity, envy and efficiency. *Journal of Economic Theory*, 9:63–91, 1974.

Ryo Yoshinaka. Higher-order matching in the linear lambda calculus in the absence of constants is NP-complete. In *Proceedings of the 16th International Conference on Term Rewriting and Applications (RTA)*, pages 235–249, 2005.